

Occupancy Grid Mapping with Highly Uncertain Range Sensors based on Inverse Particle Filters

Timo Korthals, Marvin Barther, Thomas Schöpping, Stefan Herbrechtsmeier and Ulrich Rückert
Cognitronics & Sensor Systems, Bielefeld University, Inspiration 1, 33619 Bielefeld, Germany

Keywords: Occupancy Grid Mapping, Inverse Sensor Model, Inverse Particle Filter, Uncertain Range Sensors.

Abstract: A huge number of techniques for detecting and mapping obstacles based on LIDAR and SONAR exist, though not taking approximative sensors with high levels of uncertainty into consideration. The proposed mapping method in this article is undertaken by detecting surfaces and approximating objects by distance using sensors with high localization ambiguity. Detection is based on an Inverse Particle Filter, which uses readings from single or multiple sensors as well as a robot's motion. This contribution describes the extension of the Sequential Importance Resampling filter to detect objects based on an analytical sensor model and embedding into Occupancy Grid Maps. The approach has been applied to the autonomous mini robot AMiRo in a distributed way. There were promising results for its low-power, low-cost proximity sensors in various real life mapping scenarios, which outperform the standard Inverse Sensor Model approach.

1 INTRODUCTION

Static obstacle detection for robotic systems is a well-known and commonly studied scientific field (Thrun, 2002; Thrun, 2003; Hähnel, 2004). It is a part of every local collision avoidance system set up to maneuver through cluttered environments. Another important application is the creation of obstacle maps while traversing an unknown area and the recognition of already known obstacles, so supporting the localization.

In nowadays robotic applications most mapping challenges are commonly faced by applying the best sensor for the job, which is a high accurate laser range finder in case of a mapping and localization task. These challenges mostly involve high setup costs and therefore low-cost sensors with high measurement uncertainties tend to get disregarded when it comes to algorithm design. However, these kind of sensors still make their way into mini and swarm robots, which are used for educational and research applications that demand low-cost and highly redundant setups. With today's developments in microelectronic technology, small sized robots can be built with all the functional properties of full sized robots but with the added benefits of affordability, economy of space, and fast setup time. These platforms offer a solution for carrying out real life robotics, besides simulation, on a large scale and in a cost efficient way (Navarro and Matia, 2013).

In an attempt to lower costs, hardware designers

tends to integrate sensors that may not fulfill today's accuracy requirements for mapping and localization. High uncertainty proximity sensors play a significant role in these cases. They are highly integrable and very cheap but are disadvantaged by their poor range and angle resolution. Despite this, they are commonly used for basic collision avoiding behaviors.

Our contribution overcomes the disadvantages of high ambiguity sensors by proposing an Inverse Particle Filter design, which directly samples from the analytical Inverse Sensor Model to retrieve a refined assumption of static objects in the robotics' surroundings. The particles themselves are then modeled as an Inverse Sensor Model for Occupancy Grid Mapping, which results can then directly be used for planning and navigation algorithms in situations that demand a global representation of the environment.

This paper is organized as follows: Section 2 highlights the different sensor techniques that are used for Occupancy Grid Mapping, where we also define the radiation based proximity sensor. In Section 3, we describe the design of our Inverse Particle Filter approach to a general proximity sensor model and the extension to an Inverse Sensor Model for an Occupancy Grid Map algorithm. Recommended implementation of our application is provided in Section 4. Finally, in Section 5 we explain and reflect on our various evaluations, and offer conclusions of our work in Section 6.

2 SENSORS FOR MAPPING

To realize Occupancy Grid Mapping, sensor readings are interpreted via the Inverse Sensor Model which allows the reasoning from the measurement to the actual distance of a given object (Thrun, 2002). Laser range finders and ultrasonic sensors are commonly used to acquire information about distance. This information can then be mapped onto the robot's environment model to build up its view of its surroundings.

2.1 Comparison of Sensors

The laser range finder is the most convenient sensor, as it can perform a range measurement by sending out a single ray of light. Via modulation or time-of-flight techniques, the distance of a single distinctive spot can be determined with high precision. While it is a state-of-the-art sensor for any range detection, there are also numerous drawbacks to it, namely high power consumption, mechanical parts, and high setup costs.

An ultrasonic based sensor measures distance by time-of-flight, sending out an ultrasonic wave, which is reflected and then detected for a second time. In the history of robotic applications, this sensor was one of the first for mapping and localization tasks, and has therefore been well studied (Thrun, 2002; Moravec and Elfes, 1985). The numerous disadvantages of the high ambiguity sensor lobes and the influences of an object's surface, distance, shape, reflectance and abrasiveness make this sensor relatively outdated for tasks today. Additionally, the sensor's large physical dimensions and price makes it useless for small and highly integrated robots (cf. Table 1).

Proximity sensors are commonly infrared based diode pairs, consisting of an emitting light source where the reflected intensity is measured over a certain period of time via a photo diode. The significant advantage of these kinds of sensors is their high integrability into electric circuits due their small housing. The small measurement range of a few decimeters is negligible due to the operational scenarios in mini- and swarm robots with a footprint less than 100 cm^3 (Navarro and Matía, 2013). On the other hand, the high ambiguity sensor lobe makes it apparently impossible to apply proximity sensors in mapping applications.

Figure 1 qualitatively shows the different sensor lobes that are used in Occupancy Grid Mapping. This top view is a hypothetical obstacle measurement in the sensor frame. For each sensor, the uncertainty of the objects localization is visualized by the standard

Table 1: Feature comparison of range sensors in robotics.

	Laser	Sonar	Proximity
Volume (cm^3)	> 175	~ 12	~ 0.01
Range (m)	> 10	< 5	< 0.5
Powerdraw (mW)	> 250	< 100	< 50
Precision (%)	~ 1	~ 10	~ 10
Price (\$)	> 100	~ 10	~ 1

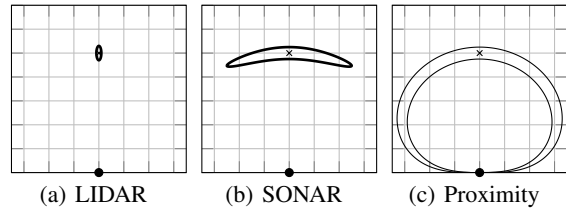


Figure 1: \bullet : Sensor position, x : Obstacle, $-$: Standard error contour of qualitative sensor cones at full range.

error contour by range and angular resolution.

2.2 Analytical Sensor Model

For our scenario, we use a radiation based proximity sensor which approximately follows the photometry inverse square law $s(x, \Theta) = \alpha_i \alpha_0 \cos \Theta / x^2 + \beta$ (Benet et al., 2002). Θ is the angle of incidence while x is the distance between sensor and object. The model's parameters can be interpreted as follows: α_0 is constant and describes the radiant intensity of the emitter, the spectral sensitivity of the receiver, and the gain of the internal amplifier. α_i depends on the object's surface and represents the reflectivity constant. β is the offset value that remains, whenever there is no object in range. The Inverse Sensor Model, s.t. determining the distance x from the measured value s is

$$s(x, \Theta) = \frac{\alpha_i \alpha_0 \cos(\Theta)}{x^2} + \beta$$

$$\Rightarrow x^2(s) \Big|_{\Theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]} = \frac{\alpha_i \alpha_0 \cos(\Theta)}{s - \beta}.$$

The disadvantage of the Inverse Sensor Model is the ambiguity of an object's position over the angle Θ and distance x (cf. Figure 1). Additionally, we introduce sensor noise σ_s which we assume to be Gaussian and constant over the angle Θ . The error propagation for the resulting distance uncertainty σ_x can be calculated as follows:

$$\sigma_x = \left. \frac{\partial x}{\partial s} \right|_{\Theta=0} \sigma_s = \frac{x}{2\alpha_i \alpha_0} \sigma_s.$$

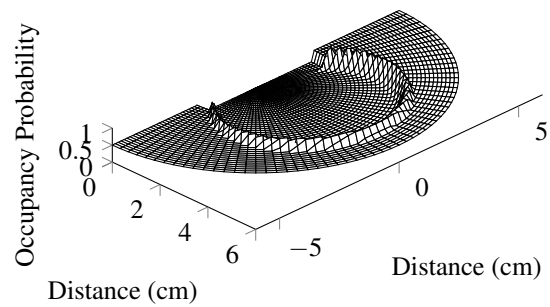
The resulting standard error of a measured object with different distances to the sensor shown qualitatively in Figure 1.

2.3 Occupancy Grid Mapping

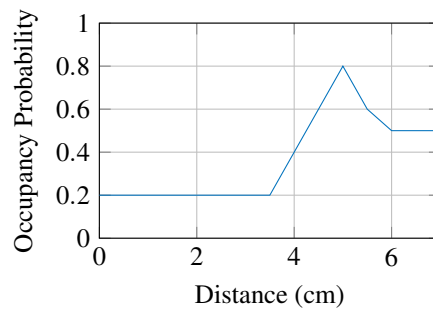
Elfes introduces the two dimensional Occupancy Grid Map (OGM) (Elfes, 1989). This representation subdivides the planar environment into a regular array of rectangular cells. Each cell represents a static location, comprising information on the covered area. Thus, the resolution of the environment representation directly depends on the size of the cells. In addition to this discretization of space, a probabilistic measure regarding state is associated with each cell. In terms of robotic navigation and mapping, this state represents space occupancy, while the mapping of other features has also been studied by the author (Korthals et al., 2015).

Hähnel states that measurements take on any real number in the interval $[0, 1]$ and describes one of two possible cell states: occupied or free. The occupancy probability of 0 means that there is definitely free space, the probability of 1 means that there is definitely occupied space and a value of 0.5 refers to an unknown state of being occupied (Hähnel, 2004). Many methods have been employed in updating the state of occupancy for each cell, such as Bayesian (Matthies and Elfes, 1988; Elfes, 1992), Dempster-Shafer (Carlson and Murphy, 2005), and Fuzzy Logic (Plascencia and Bendtsen, 2009).

A probabilistic Inverse Sensor Model (ISM) is used to update the OGM in a Bayesian framework, which deduces the occupancy probability of a cell, given its sensor measurement. The technique to derive models for LIDAR and SONAR sensors have been well studied by Thrun (Thrun et al., 2005), Hähnel (Hähnel, 2004), and Stachniss (Stachniss, 2009). We adopted the ISM from Stachniss and Hähnel for proximity sensors as depicted in Figure 2.

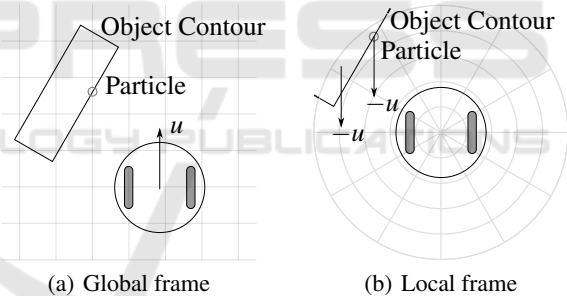


(a) Planar occupancy probability



(b) Occupancy probability along radial axis

Figure 2: Inverse Sensor Model for radiation based proximity sensors measuring an object at 5 cm.



(a) Global frame

(b) Local frame

Figure 3: Locomotion of robot and objects for steering command u .

3 INVERTED PARTICLE FILTER

Particle filters are commonly used to find the position of a robot based on measurements of its environment. The Inverse Particle Filter (IPF) described in this section inverts the problem: It finds the position of the environment based on measurements from its position. The output from the IPF is therefore an estimate, based on each particle, of the surfaces of objects surrounding the robot.

In order to update the IPF and global OGM, the robot needs two representations of its environment: First, the IPF resides in a local circular map with the robot in its center. Second, the OGM is a global map with a fixed coordinate system in space, which can be traversed by the robot. As the robot moves, each

particle in the local map will move in the opposite direction respectively. This reflects the position of the particle relative to the new position of the robot, as shown in Figure 3(b). In the OGM frame, shown in Figure 3(a), the situation is reversed, where each sampled particle remains in position and represents a static object in space.

For our initial work, we used the IPF technique approach as proposed by Kleppe (Kleppe and Skavhaug, 2013). Kleppe applied the IPF to a single laser range finding application that was mounted on a mini robot with restricted movement. In the following sections, we accommodate his work and extend the IPF in three ways: First, we use a full analytical model of the sensor instead of heuristics to make the IPF generally valid. Second, the extension to a multi sensor setup

Table 2: Variables used in the SIR filter.

I	Number of particles
$\{\cdot\}_{i=1}^I$	Feature set of I particles
k	Particle position in local frame
t	Time step
w	Weight of a particle
e	Epoch counter
z	Sensor measurement
u	Steering command
$p(p k, u)$	Error model of the inverse kinematic
$p(z k)$	PDF of the analytical ISM

is proposed and third, we finalize our IPF with an underlying OGM, so that our approach can be directly fed into any common robotic navigation application.

3.1 Inverse Particle Filter

To accomplish the objects' localization from the readings, we must use the Sequential Importance Resampling (SIR) filter as described in Algorithm 1 with the variables from Table 2. First, the Inverse Kinematic Model (IKM) is applied to the old set $\{\cdot\}_{i=1}^I$ of particles. The IKM is the relative movement of the particles in the robots frame with the respective error model. A detailed description is given in Section 4.4 for a robot with differential kinematic. Second, the score of each particle, that being an objects surface representation, is calculated by evaluating the new ISM at each particle position.

Algorithm 1: SIR-Filter.

Require: $\{k_{t-1}, w_{t-1}, e_{t-1}\}_{i=1}^I, z_t, u_t$

- 1: **for all** k_{t-1} in range **do**
- 2: $\tilde{p}_t^{(i)} \sim p(p_t|k_{t-1}^{(i)}, u_t)$ ▷ Apply IKM
- 3: $\tilde{w}_t^{(i)} = p(z_t|k_t^{(i)})$ ▷ Apply ISM
- 4: **end for**
- 5: **for all** \tilde{w}_t **do**
- 6: $w_t = \text{Normalize}(\tilde{w}_t)$
- 7: **end for**
- 8: **for all** k_t **do**
- 9: $k_t = \text{Resample}(w_t)$
- 10: $e_t = e_t + 1$
- 11: **end for**

Ensure: $\{k_t, w_t\}_{i=1}^I$

A new aspect in our SIR filter can be found by counting the particles' survived epochs e . This value is used for Innovation Gating (IG) in the OGM, where only particles are taken as true obstacle estimations if the epoch counter is greater than a certain gating value.

To ensure fast convergence, we do not generate

new samples randomly, but directly from the ISM via Algorithm 2. If we want to find the hidden distribution of obstacles in the environment, then ISM gives us the best approximation of the unknown distribution. N denotes the number of sensors attached to the robot, and s_t the measured value of sensor n at time t .

Algorithm 2: Generation of new Particles.

Require: $\{s_t\}_{n=1}^N$

- 1: $\tilde{n} \sim \mathcal{U}(1, N)$ ▷ Draw a sensor
- 2: $\tilde{\phi} \sim \mathcal{U}(-90^\circ, 90^\circ)$ ▷ Draw a local angle
- 3: $\tilde{r} \sim \mathcal{N}(s_{\tilde{n}}, \sigma_{s_{\tilde{n}}})$ ▷ Draw a local distance to the sensor
- 4: $x_t = \{\tilde{r}, \tilde{\phi}\}$

Ensure: k_t

An example of two IPF steps can be seen in Figure 4. At $t = 0$, an initial measurement of an object that is 10 cm in front of the sensor is received. Thus, the IPF is bootstrapped, drawing 100 % of particles, s.t. object hypothesis, from the sensor model. Next, the sensor moves towards the object which results in the relative motion for all particles and the object in the sensor frame. Here, the IKM is applied to every particle. In Figure 4(c), a score is calculated for every particle based on the current measurement. 80 % of particles survive this epoch, while the remaining 20 % are rejected and sampled from the current sensor model. At $t = 2$, a consecutive motion and importance resampling is applied again.

The movement u_t is a very significant feature for acquiring the objects' positions from the ambiguous sensor readings z_t , due to the fact that the history of sensor readings is strongly correlated to the real objects' positions. The movement and error propagation of the particle is derived for a differential kinematic in Section 4.4. It also affects the sampling frequency of the IPF itself so that we introduce a sample-per-speed ratio SSR with

$$[SSR] = \left[\frac{\text{IPF Sample Frequency}}{\text{Linear Vehicle Velocity}} \right] = \frac{\text{Hz}}{\text{m/s}} \quad (1)$$

which has to remain constant at any time to ensure functionality of the IPF. If this cannot be guaranteed, e.g., if the robot stands still, the IPF needs to be temporarily switched off. Otherwise, it would find random objects based on sensor noise.

To respect more than one sensor via the IPF, we can extend the calculation to weights of each particle. The overall weight $w_{t,N}$ results from evaluating each ISM of N sensors at the particles position in the local frame. The derived probabilities are then fused by De Morgan's law as follows:

$$w_{t,N} = P(x_t|s_t) = 1 - \prod_{n=1}^N (1 - P(x_t|s_{t,n})). \quad (2)$$

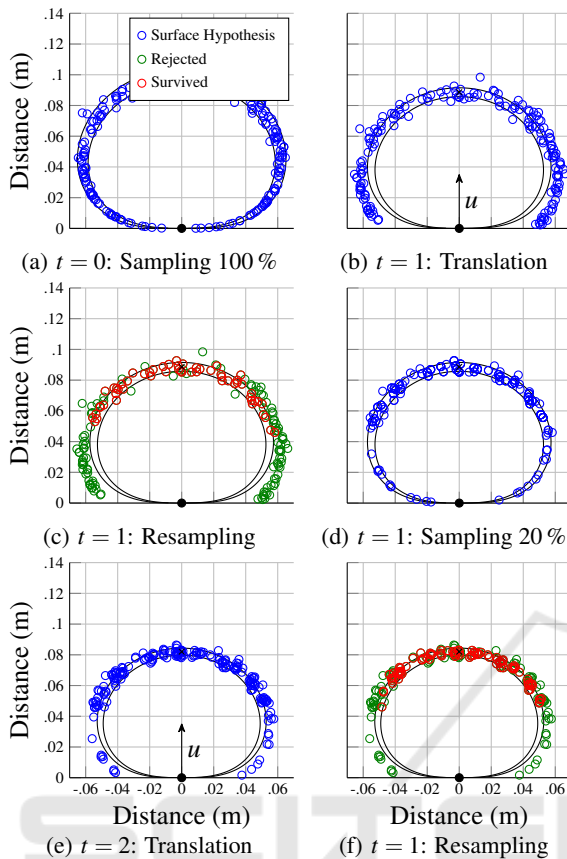


Figure 4: Two steps of the SIR-Filter depicted line-wise from top-left to bottom-right in the sensor frame. •: Sensor position, x: Obstacle, -: Standard error contour.

3.2 Learning Occupancy Grid Maps

Our new approach updates a cell $c = \{c_{\text{free}}, c_{\text{occ}}\}$ in the map by interpreting each particle falling within this cell as follows:

$$\begin{aligned} & c_{\text{free}} + 1 && \text{, if } i \text{ was rejected,} \\ & c_{\text{occ}} + 1 && \text{, if } i \text{ survived and } e_i \geq g, \\ & \text{back off} && \text{, if } i \text{ survived and } e_i < g. \end{aligned} \quad (3)$$

Every resampled particle k_i that was rejected for the next epoch is taken as free space since it does not match the surface estimate of a wall. On the other hand, all sampled particles surviving the epoch are interpreted as occupied, taking IG into account. Particles that haven't survived the necessary amount of epochs are backed off and do not increase c_{free} nor c_{occ} . Finally, the state of a cell is calculated by applying the counting model (Hähnel, 2004):

$$f_{\text{occ}}(c_{\text{occ}}, c_{\text{free}}) = \frac{c_{\text{occ}}}{c_{\text{occ}} + c_{\text{free}}}. \quad (4)$$

Figure 5 shows the result of sorting the particles (cf. Figure 5(a)) from the steps depicted in Figure 4

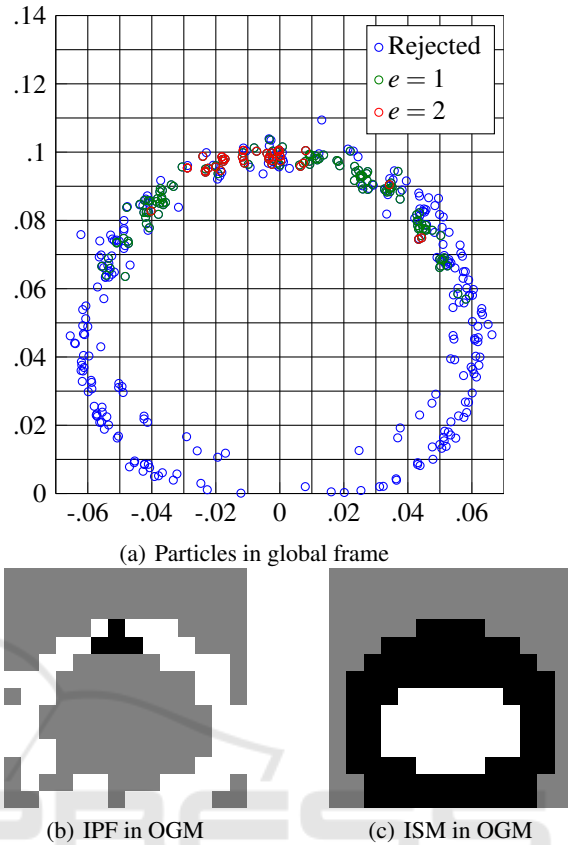


Figure 5: OGM comparison between IPF and ISM.

into an OGM (cf. Figure 5(b)). The OGM reference frame is a fixed global one and coincides with the initial sensor position at $t = 0$. Figure 5(a) shows the survived and rejected particles in the initial sensor frame at $t = 0$ with an overlaid discretization. The corresponding OGM is deduced by applying the rules from Equation 3 and Equation 4 with $g = 2$. Figure 5(c) shows the same OGM but created by the standard ISM derived from the analytical sensor model.

4 IMPLEMENTATION

We implemented and tested our Inverse Sensor Model approach on the Autonomous Mini Robot (AMiRo). Section 4.1 briefly describes the robot itself and its hardware. Our approach is depicted in Section 4.2. Finally, the analytical sensor model for the robots proximity sensors is analyzed and optimized in Section 4.3.

4.1 AMiRo

As can be seen in Figure 6(a), AMiRo was developed

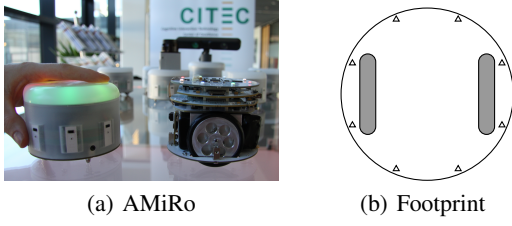


Figure 6: AMiRo platform with footprint of kinematic and its eight co-circular arranged proximity sensors (Δ).

by Herbrechtmeier et. al. (Herbrechtsmeier et al., 2012) at the Cognitive Interaction Technology - Center of Excellence (University of Bielefeld, Germany). The robot has a cylindrical shape with a diameter of 100 mm and a height of 76 mm. Two wheels give the ability for locomotion through differential steering. Our setup of the AMiRo consists of three different modules, each of which has been designed to handle specific tasks: *DiWheelDrive* (STM32F103 MCU for motion control and odometry), *PowerManagement* (STM32F405 MCU for object recognition, and basic behavior), *Cognition* (DM3730 SoC for high-level planning tasks and WLAN communication). In order to permit the mapping task, *PowerManagement* is extended by eight integrated sensors, arranged as depicted in Figure 6(b) for proximity measurements.

4.2 Realization

Our approach is implemented on the AMs in a distributed fashion. The IPF, s.t. the periodic sampling of distant surfaces, strongly demands a real-time approach to ensure that all samples and particle weights are calculated before a new sensor measurement arrives. Thus, this part was carried out on the *PowerManagement* AM. Another important aspect of our approach is the fact, that the quality of sampling strongly depended on driving distance (see Section 3). This is satisfied by keeping the *SSR* constant via supersampling the sensors at 125 Hz and downsampling to any requested rate.

The mapping, s.t. the sorting of the particles into an array structure, is a highly memory consuming application with respect to the limited hardware resources on the robot. Though not mandatory, it also requires dynamic allocatable memory to extend map dimensions on demand. Therefore, it is unfeasible for it to be realized in a real-time fashion on the MCU based boards and thus become implemented on the *Cognition* AM.

For evaluation, the *Cognition* AM logs all steering, localization and sensor messages. Afterwards, we analyze, evaluate, and prove our approach via a MATLAB based implementation.

4.3 Optimized Sensor Model

For our proximity sensor, we chose the integrated infrared based proximity sensors VCNL4020 by Vishay, which only approximately follows the photometry inverse square law as described by Benet et al. (Benet et al., 2002). The maximum range of the sensors x_{\max} is about 20 cm, while the maximum angle of incidence θ_{\max} is at 60° .

For the correction of the cone characteristic the constant ξ has been added to modulate the cosine. Additionally, the constant α depends on the distance of the object. This behavior originates from the fact that if the object were to be closer to the sensor, the reflecting ground area would also decrease. Thus, the constant α is extended by a linear function in dependence of its sensor value. Finally the advanced sensor model is defined as:

$$x^2(s, \Theta) = \frac{\alpha_i \alpha_0 \cos(\xi \Theta)}{s - \beta} + \delta \cos(\xi \Theta),$$

$$\sigma_x(x, \Theta) = \frac{\left(\frac{x^2}{\cos(\xi \Theta)} - \delta \right)^2}{\frac{2}{\sqrt{\cos(\xi \Theta)}} \alpha_i \alpha_0 x} \sigma_s. \quad (5)$$

The ISM $x(s, \Theta)$ and error model $\sigma_x(x, \Theta)$ from Equation 5 are used by the IPF in our evaluation scenario.

4.4 Inverse Kinematic Model

Since the IPF resides in the robot frame, the robot does not move by the steering command u , but rather each particle respectively. The vehicles velocities (v_x, v_y, ω) in the global planar frame are determined using a differential kinematic's steering commands $u = (v_l, v_r)$, as illustrated in Figure 3, and the basewidth b after the Euler method as shown in Equation 6. Further, we can omit the robot's orientation Θ since we are only interested in the particles' velocities in each step in the robot's frame

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \stackrel{\Theta=0}{=} \begin{bmatrix} 1/2(v_l + v_r) \\ 0 \\ 1/b(v_l - v_r) \end{bmatrix}. \quad (6)$$

From Equation 6, the relative particle motion denoted by "P" in the vehicle frame at position (r, θ) can be written as follows:

$$\begin{bmatrix} v_{P,x} \\ v_{P,y} \end{bmatrix} = \begin{bmatrix} \frac{v_l + v_r}{2} - \frac{v_l - v_r}{b} r \sin(\phi) \\ -\frac{v_l - v_r}{b} r \cos(\phi) \end{bmatrix}. \quad (7)$$

The particle's next position, F , can now be directly derived from Equation 7:

$$F = \begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + \Delta t \cdot v_{P,x} \\ y_t + \Delta t \cdot v_{P,y} \end{bmatrix}. \quad (8)$$

Assuming Gaussian uncertainty for each particle, one can now derive error propagation from Equation 8 by calculating the Jacobian J_F of F . The resulting model can be directly applied as IKM in Algorithm 1.

5 EVALUATION

We evaluated various mapping scenarios to benchmark our approach in the individual situations (cf. Figure 7). All objects consist out of acrylic blocks with the dimensions 200 mm x 50 mm x 70 mm. Therefore, we decided that every object construction should be at least of the same dimensions of the robot. The robot drives with 80 mm/s, while the sensor frequency is 8 Hz. Thus, by applying Equation 1, we gain a *SSR* of 0.1 Hz per mm/s.

Quantitative benchmarking is carried out by evaluating the resulting OGM after path traversal against a deduced map by an adapted ISM for sonar sensors, as described in Section 2.3. We calculate the Overall Mapping Errors (OE) with respect to the Ground Truth (GT) map as proposed by Carvalho (Carvalho and Ventura, 2013):

$$OE = \frac{\# \text{ False Positives} + \# \text{ False Negatives}}{\# \text{ All Classified Cells}}. \quad (9)$$

For benchmarking, we decided to expand the walls by a maximum of half the robots radius for benchmarking. This was done because measurements of a proximity sensor are not absolutely accurate, as the name suggests. But common navigation algorithms for OGMs expand all found walls by at least half of the robots radius anyway. Thus, we take the best OE value within this range.

The chosen scores do not fully comprise the quality of the deduced OGMs. It is not really beneficial compare the standard ISM against the IPF of the full driven paths, as it is not known how the algorithms will perform in between. This is important for navigation and obstacle avoidance algorithms that must perform the path traversal on the OGM.

5.1 Scenario Description

In the **Single Sensor - Single Object (SSSO)** scenario, a single block is placed horizontally in front of a sensor. The robot approaches the short side of the block. It starts at a distance of 350 mm to the block and approaches it until it is 60 mm away. The **Multiple Sensors - Single Object (MSSO)** scenario is the same as the SSSO, but with all eight sensors activated.

For the **Multiple Sensors - Object Drive-by (MSOD)** scenario, a single block is placed horizontally beside the robot. The robot drives along the

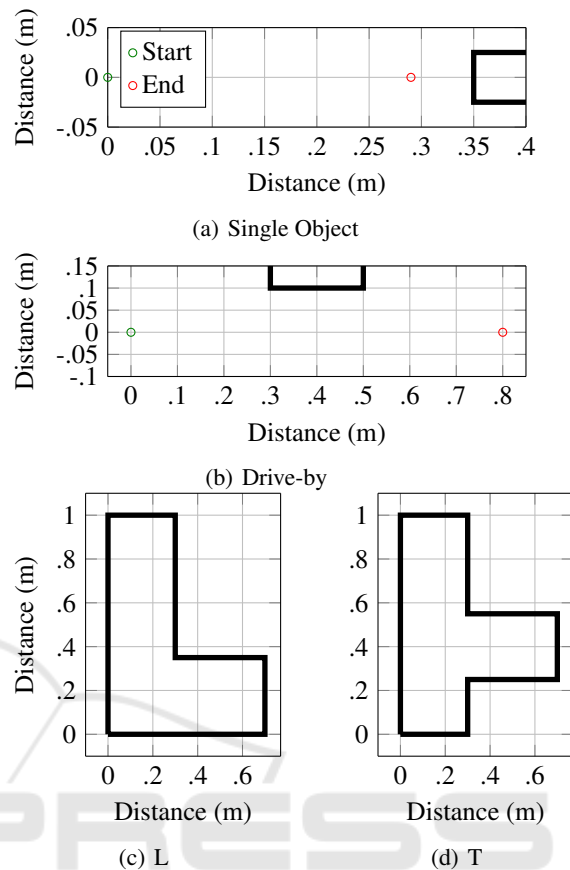


Figure 7: Ground Truth scenarios.

long side of the block at a constant surface distance of 100 mm with all sensors activated. In the **Multiple Sensors - Maze (MSM)** scenarios, multiple blocks are arranged in L- and T-shape mazes. Within the navigation task, the robot is required to drive on a pre-defined trajectory with all sensors activated.

5.2 Benchmark

Figure 9 shows different OGM mappings of the ISM in comparison to the new proposed IPF. The OGMs are visualized, such that white space denotes values $f_{occ} < 0.5$, black space $f_{occ} > 0.5$, and grey space (s.t. unknown, with no readings acquired yet) $f_{occ} = 0.5$. The red shape is the obstacles GT overlay. All IPF related OGMs are deduced with $g = 4$ and 1000 particles per sensor.

Figure 9(a) and Figure 9(b) show the real outcome for the artificial scenario depicted in Figure 5. For the ISM, the contour is no longer closed due to the free space readings along the path, which override these values. It is clear that the ISM produces an obstacle in the OGM, which itself has the concave shape of the sensor cone. The IPF, in contrast, shows a far superior

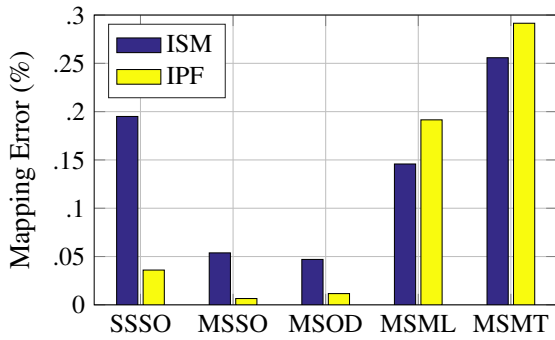


Figure 8: Overall Mapping Errors for depicted scenarios.

obstacle recognition so that the unrelated sensor cone parts of the obstacle readings can be correctly rejected (cf. Figure 9(e)). Therefore, a more detailed shape will remain.

In the multi sensor case, the ISM is depicted without (cf. Figure 9(c)) and with (cf. Figure 9(d)) the maximal range readings (cf. (Thrun, 2002)). It seems clear, that these values need to be filtered and handled separately, such that maximal range readings are first cut and then drawn in as free space. This results in a significant advantage of the IPF over the ISM such that raw sensor readings can be directly fed into the IPF. The range sensitivity is almost doubled and thus, the shape of the object is also deduced with higher similarity. The adjacent sensors also rectify the object, which results in erroneous deduction at the object's sides. Again, the IPF filters these readings and then marks them as free, neither is this correct nor does it encumber the navigation algorithms that use the OGM. The MSOD scenario confirms the superiority of the IPF (cf. Figure 9(f) and Figure 9(g)). Not only are the range readings longer but also the objects shape is better deduced.

Visually, the mappings hardly differ in the T- and L-shape scenarios (cf. Figure 9(h) to Figure 9(k)). Here, the ISM performs better on first sight, without erroneous readings close to the walls. But the real drawback becomes apparent once the maps that have been built during path traversal are compared (cf. Figure 9(l) and Figure 9(m)). The maze presents such a cluttered environment, that all sensors have continuous object readings. This makes the OGM built by ISM unusable for any navigation algorithm. The IPFs behavior is again superior, deducing the ongoing area as free.

Quantitatively speaking, the OE in Figure 8 approves the visual inspection. In all free space scenarios, the IPF significantly outperforms the ISM. It is only slightly outstripped in the maze scenarios, which is negligible due to the former discussion.

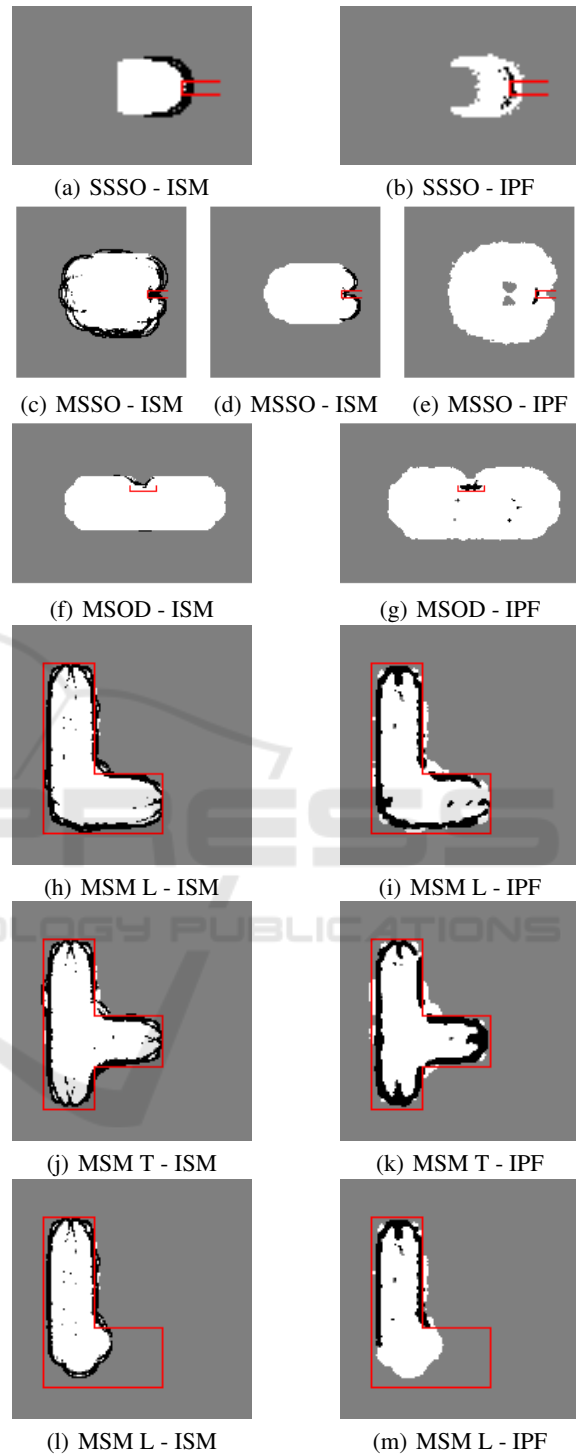


Figure 9: OGMs deduced by ISM and IPF.

6 CONCLUSION

This paper presents a novel approach for Occupancy Grid Mapping, using sensors with low range and angle resolution compared to the more commonly used sensors in mapping tasks. The solution is based on a sequential Monte Carlo method named the Inverse Particle Filter, as it reverses the localization problem. The paper fully describes how to design the algorithm, and has shown a functional application on the microcontroller based robot AMiRo in a distributed fashion. The Inverse Particle Filter significantly outperforms the standard ISM approach in a quantitative way and visually, so that common navigation algorithms in both free space and cluttered environments can directly use the deduced Occupancy Grid Maps. Further research concentrates on the comparison of the Inverse Particle Filter for arbitrary sensors and inverse sensor models, in order to overcome the Occupancy Grid Map's functional lack of heterogeneous sensor setups. Additionally, the Inverse Particle Filter has been planned to respect non-static environments which makes our approach fully suitable for Bayesian occupancy filtering.

ACKNOWLEDGEMENTS

This research/work was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster "Intelligent Technical Systems OstWestfalenLippe" (it's OWL) and managed by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the contents of this publication.

REFERENCES

Benet, G., Blanes, F., Simó, J., and Pérez, P. (2002). Using infrared sensors for distance measurement in mobile robots. *Robotics and Autonomous Systems*, 40(4):255–266.

Carlson, J. and Murphy, R. (2005). Use of Dempster-Shafer Conflict Metric to Detect Interpretation Inconsistency. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005)*, abs/1207.1.

Carvalho, J. and Ventura, R. (2013). Comparative evaluation of occupancy grid mapping methods using sonar

sensors. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7887 LNCS:889–896.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57.

Elfes, A. (1992). Dynamic control of robot perception using multi-property inference grids.

Hähnel, D. (2004). *Mapping with Mobile Robots*. PhD thesis.

Herbrechtsmeier, S., Rückert, U., and Sitte, J. (2012). *AMiRo - Autonomous Mini Robot for research and education*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kleppe, A. L. and Skavhaug, A. (2013). Obstacle detection and mapping in low-cost, low-power multi-robot systems using an Inverted Particle Filter.

Korthals, T., Krause, T., and Rückert, U. (2015). Evidence Grid Based Information Fusion for Semantic Classifiers in Dynamic Sensor Networks. *Machine Learning for Cyber Physical Systems*, 1(1):6.

Matthies, L. and Elfes, A. (1988). Integration of sonar and stereo range data using a grid-based representation. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 727–733. IEEE.

Moravec, H. and Elfes, a. (1985). High resolution maps from wide angle sonar. *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, 2.

Navarro, I. and Matfa, F. (2013). An Introduction to Swarm Robotics. *ISRN Robotics*, 2013:1–10.

Plascencia, A. C. and Bendtsen, J. D. (2009). Sensor Fusion Map Building-Based on Fuzzy Logic Using Sonar and SIFT Measurements. In *Applications of Soft Computing*, pages 13–22. Springer.

Stachniss, C. (2009). *Robotic Mapping and Exploration*.

Thrun, S. (2002). Robotic Mapping: A Survey. (February).

Thrun, S. (2003). Learning occupancy grid maps with forward sensor models. *Autonomous Robots*, 15(2):111–127.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*.